2

Interim Report

ORSER-SSEL Technical Report 10-73

PROGRAM DESCRIPTIONS

F. Y. Borden, D. N. Applegate, B. J. Turner,
H. M. Lachowski, and J. R. Hoosty

ERTS Investigation 082
Contract Number NAS 5-23133

INTERDISCIPLINARY APPLICATION AND INTERPRETATION OF ERTS DATA WITHIN THE SUSQUEHANNA RIVER BASIN

Resource Inventory, Land Use, and Pollution

Office for Remote Sensing of Earth Resources (ORSER)
Space Science and Engineering Laboratory (SSEL)
Room 219 Electrical Engineering West
The Pennsylvania State University
University Park, Pennsylvania 16802

Principal Investigators:

Dr. George J. McMurtry
Dr. Gary W. Petersen

Date: May 1973

ORSER-SSEL
Technical Report 10-73
May 1973

P R O G R A M D E S C R I P T I O N S

F. Y. Borden, D. N. Applegate, B. J. Turner,

H. M. Lachowski, and J. R. Hoosty

Office for Remote Sensing of Earth Resources
A Division of the Space Science and
Engineering Laboratory
The Pennsylvania State University
University Park, Pennsylvania

CONTENTS

DIGITAL AIRCRAFT MULTISPECTRAL SCANNER DATA TAPE FORMAT

F. Y. Borden

The format herein described was designed for any multispectral scanner data collected from an airborne platform. All programs written for the Office for Remote Sensing of Earth Resources (ORSER) at The Pennsylvania State University for processing this kind of data will accept this format. Digital tapes with other formats, such as the "Aircraft Data Storage Tape Format" of LARSYS Version 2, can be reformatted to agree with these specifications without serious difficulty.

Data Organization

Data on the tape is organized according to flightlines, with each flightline of data comprising a unique file. The full set or a subset of the full set of data for a flightline consists of a group of physical and logical tape records. One or more flightline data sets can be stored on one tape and an incomplete set can be continued from one tape to another.

Each file of flightline data is composed of five kinds of records as follows and in the order given:

1. File identification - 1 or more records, each 250 words in length.
2. Table of contents - 1 record of 400 words in length.
3. Multispectral scanner response records - 1 or more records per scan line containing all channel responses for all scan line elements.
4. History record sets - 1 set of records for each main program execution which caused the file to be modified.
5. End of file record - 1 record 250 words in length.

File Identification Record

The file identification record contains the information relating to the data set for the flightline. It should agree with the external documentary information for the flightline. The record is fixed in length at 250 words.

Word	Format	Contents
1-3	Alpha	Flightline or user ID (12 characters)
4	Integer	Continuation code 0 - No ID records following first one n - n ID records following first one
5	"	Number of data channels
6	"	Original number of elements per scan line
7-10	Alpha	ORSER external tape label (16 characters)
11	Integer	Month data were collected
12	"	Day data were collected
13	"	Year data were collected
14	"	Time of day data were collected
15	"	Altitude above ground of aircraft
16	"	Ground heading of aircraft

17-19	Alpha	Date this specific data set was prepared (12 characters)
20	Integer	Air speed (mph)
21	Alpha	Type of original tape: ERTS, C130, U2, LARS
22-25	Alpha	Platform description
26-30	Alpha	Scanner description
31	Integer	Milliradians per element 1 = present 0 = absent
33-37	Alpha	Name of user who created this data set
38-41	Alpha	ORSER external label of subset source tape
42	Integer	Subset source tape file number
43	Integer	File number of this tape
44-50		Unused
51	Integer	Number of first spectral band (channel) on file
52	Real	Lower limits in micrometers of first spectral band in file
53	"	Upper limits in micrometers of first spectral band in file
54	"	0 or suggested value of C_0 calibration pulse

55	Real	0 or suggested value of C_1 calibration pulse
56	"	0 or suggested value of C_2 calibration pulse
57-199	"	Repetition of description for words 51-56 applied to other channels in file in order of appearance in data
200-250	Alpha	ERTS ID record if this tape was generated from a NASA-ERTS tape

If the tape has been generated from C130, U2, or LARS aircraft data, a second ID record will be present that will contain the original ID record from the original tape.

Table of Contents Record

The table of contents record contains the list of all data blocks in the file. A data block is defined as all data from a beginning scan line through an ending scan line including all elements in each scan line beginning with a given element number and ending with a given element number. As many as 50 different blocks can exist in the file. The table of contents is a 400-word fixed-length record composed of 50 eight-word sets. Each non-zero set applies to one of the blocks in the file. The specifications for the first set are as follows:

Word	Format	Contents
1	Integer	Beginning scan line number for the block
51	"	Ending scan line number for the block
101	"	Beginning element number for each line in the block
151	"	Ending element number for each line in the block
201	"	Increment for scan line numbers in the block; i.e., an increment of 1 means every line is present, whereas an increment of 3 means every third line is present
251	"	Increment for element numbers in all scan lines of the block
301	"	Number of scan lines in the block
351	"	Number of elements in a scan line

Multispectral Scanner Response Records

One or more records exist for each scan line of data and include all selected elements and all channels for that scan line. If the number of elements per scan line is 222 or less, and the number of channels is 13 or less, one record contains all data for the scan line. Otherwise, the

additional data is contained in the same format on continuation records.

Byte	Contents
1-2	Scan line number
3-4	Roll parameter
5-6	Beginning element number in the line
7-8	Ending element number in the line
9-10	Element number increment
11-12	Continuation code
13-n[1]	Responses ordered as follows:
	first channel, first element
	first channel, second element
	. .
	. .
	. .
	first channel, last element
	calibration data for first channel (8 bytes)
	second channel, first element
	second channel, second element
	. .
	. .
	. .
	last channel, first element

[1]n = m * (e + 8) + 12 for m number of channels and e number of elements for the line, n $\leq$ 2976.

	last channel, second element
	. .
	. .
	. .
	last channel, last element
n-7 to n	calibration data for last channel (8 bytes)

History Records

Each step in the data processing that generates a modified file will cause a set of history records to be added to the modified file. This set will be added to any prior history records.

History Header Record. The first record of the set is an eight-word fixed-length record. The format of the header record is as follows:

Word	Format	Contents
1-4	Alpha	Name of program that made data set modification (16 characters)
5-6	"	Run date
7	Integer	Run identification
8	"	Number of records following this one for this set (n)

History Data Records. Following each header record n, the value in word eight of the history record, records with an 80A1 format follow. These records are typically the control card images for the run.

End of File Record

The end of file record is fixed in length at 250 words and is a reproduction of the identification record of the file.

Data Subsets

For computer use cost economy, it is desirable to minimize tape processing time. For this reason it is anticipated that a user will construct one or more subsets of data to delete sections that are of no interest to him. Data to be used frequently in processing would be selected in order to omit the long tape search time required if the data had to be selected from the full set each time. To do this the user selects blocks of data he is interested in and prepares a tape containing only those blocks. All subset data tapes have this same format and all programs have been

designed to accept this format whether from an original tape or from any subset thereof.

The table of contents, as well as the history records, is valuable for this reason from the point of view of control of the data for the user. Use of the program to construct data subsets is described elsewhere, but the programs can be used to construct smaller subsets of data from a subset data tape. Other programs, for example sampling programs, also can be expected to modify data sets. The necessity and value of the table of contents and the history records are therefore clear.

TPINFO Program Description

H. M. Lachowski

The primary purpose of TPINFO is to output information for an original or any SUBSET data tape containing digitized aircraft multispectral scanner data. The information of interest to the user is contained in the Identification (ID) record and in the Table of Contents records at the beginning of each tape. Detailed descriptions of these records may be found in the Digital Aircraft Multispectral Scanner Data Format documentation.

By using control cards, the user may request the following output from the TPINFO program:

1. ID record,
2. Table of Contents record,
3. Response record (the first record following the Table of Contents), and
4. HISTORY records.

The TPINFO program is intended mainly for the user who is not sure about the flightline name, certain parameters

such as the number of observations per scan line, the number of data channels, or what blocks of data are contained on a given tape file. In most cases, the ID and the Table of Contents records will be sufficient.

SUBSET PROGRAM DESCRIPTION

F. Y. Borden and H. M. Lachowski

The SUBSET program is used primarily to increase tape processing efficiency and to reduce computation cost. A flightline generally results in a large digital file frequently consisting of more than one full tape reel. A user is usually interested in only relatively small parts of a flightline. The SUBSET program allows the user to specify the parts of a flightline he is interested in and to construct onto his tape a subset of the data that contains only the data he specifies. Once the subset data tape has been constructed, subsequent processing using this tape avoids the costly bypassing of unwanted data as would be the case if the original tape were to be used. The SUBSET program can also be used to select a smaller subset from a tape that has been constructed as a subset of the original or other prior subset of data.

A typical example of the use of SUBSET follows. Suppose a user desires to study certain soil areas that he can identify in a general way from aerial photography associated with a flightline. SUBSET can be used to construct a subset of the flightline data containing only the soil areas to be investigated. Once this tape has been prepared, the user may want to select a number of smaller areas that contain the training areas to be used to develop classification parameters for statistical classification procedures. This could be done by constructing another subset of the data using SUBSET with the first subset data tape used as input. When the classification parameters have been estimated to the user's satisfaction using the second subset of the data, the classification could be run on the first subset. In this way a minimum amount of unused data would have to be passed in each computer run and a substantial cost savings would result compared to using the original full flightline of data for each run.

Tape Formats and Data Organization

Every subset of data output by SUBSET has the same tape format as the original data tape. Every subset tape can be processed by any of the programs that can operate on the original data tape. The detailed description of the tape format is given in the "Digital Aircraft Multispectral Scanner Data Tape Format" manual. The identification record of the source tape is reproduced on the subset tape with only the name of the tape changed to the name specified by the user. The table of contents record for the subset tape is constructed from the input specifications of the user and the table of contents from the input tape. The table of contents specifies exactly the contents of the subset tape. The data on every subset tape is always in the order of increasing scan line number. There are no duplications of line numbers and no out-of-order scan lines. The section describing block restructuring presents more details regarding the record organization for the scan lines. The history records are reproduced onto the subset tape and augmented appropriately for the run. Any inputting

subroutines that work for the original tape, such as GETLIN, will function properly for any subset data tape.

Specifying a Subset

A subset of data is defined by specifying one or more blocks of data that will be selected from the source set to be put on the subset tape. A block is composed of data beginning with a designated scan line and ending with a designated line and including elements in each line from a beginning element number through an ending element number. This is comparable to a rectangular area along the flightline with two sides parallel to the flightline. Not all lines are necessarily included in a block; a line sampling increment can be used to prescribe the spacing between lines to be selected. In the same way, not all elements within the limits need to be selected since an element sampling increment can be prescribed. In specifying blocks, the source data tape table of contents must be consulted to insure that the source from which the subset is chosen contains the desired lines and elements within lines. A number of inconsistencies between the requested block

delimiters and the actual data available on the source tape are allowed and adjustments are automatically made in the block requests. However, to insure a completely satisfactory subset, the data available, as given in the table of contents of the source tape, should be used as a reference in specifying blocks for the subset. The rules governing the adjustments for inconsistencies are covered in a later section.

It is expected that blocks in the requested subset will frequently overlap. Two types of overlap can occur. First, blocks can spatially overlap in that one may be wholly or partly included in one or more others. Second, blocks may overlap by having some or all scan lines in common, but not be spatially overlapping. Since the subset tape will contain no duplicate line numbers and will contain all elements in a scan line within only two delimiters, the requested blocks in overlapping cases will be restructured internally. The exact nature of the restructuring is described in a later section. However, the SUBSET program will cause all of the required data to be present in the subset if the requests are consistent with the available set. For each inconsistency, a reasonable substitution will be made if possible; otherwise, the requested block will be overlooked. Output messages fully cover the unusual

situations. In addition, in every run the requested table of contents and an actual table of contents for the output subset are printed. SUBSET, in all cases, will generate the minimum subset of data that includes all of the requested data (considering substitutions or deletions as above) within the constraints of the data tape format and organization.

Block Restructuring

Whenever requested blocks of data overlap, the overlapping blocks are restructured automatically into non-overlapping blocks. None of the requested data will be lost in the process, but, frequently, additional data will be included as a result of the restructuring. An additional kind of overlapping brings about restructuring; i.e., a requested block that does not fall completely within a block is partitioned into two parts, the first of which does fall completely within one of the input tape blocks. The second partition is put back in the table of requests and is considered as a separate block later when its turn comes in the run.

Restructuring is necessary where overlapping occurs to avoid duplication of scan line numbers in the subset tape which, in turn, circumvents complicated programing and processing for subset tapes. On the other hand, automatic restructuring releases the user from undue restrictions or complications in specifying the blocks he wants.

In the restructuring, overlapping blocks are partitioned and recombined. Where recombination takes place, the smallest line increment and element increment of the partitions of the original requested blocks apply. The table of contents is changed to agree exactly with the restructuring.

The rules for restructuring two overlapping blocks, block i and block i-1 are given below and illustrated in Figures 1 through 5. The overlapping condition is given in the diagrams on the left side of the arrow and the result of the restructuring is given on the right. The smallest scan line for a block is at the bottom of the rectangle. The tests for the various conditions are applied repetitiously for each block i and restructuring is made when necessary until, finally, no overlapping remains.

If one block is completely included in another and the element increments are the same, the included block is

deleted as a separate entity (Figure 1). If the element increments are not the same, the blocks are reconstructed.

For two blocks in which the scan lines overlap and do not begin on the same line, the block with the lowest numbered scan line is partitioned. The first part ends one line before the beginning line of the overlapping block and the second part covers the remainder, beginning at the same line as for the overlapping block (see Figures 2 and 3).

For a second block that is a continuation of a first block, the two are combined when the line and element increments match (Figure 4).

For two blocks in which the scan lines overlap and have the same beginning scan line, the blocks are restructured so that one includes all of one of the original blocks and part of the other and the other restructured block contains what is left over (Figure 5). In Figure 5 the data within the dashed (*) lines is included although it was not requested.

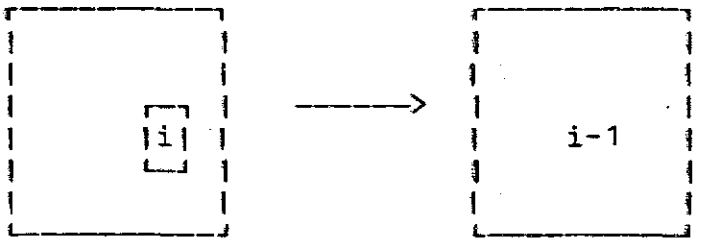

Figure 1

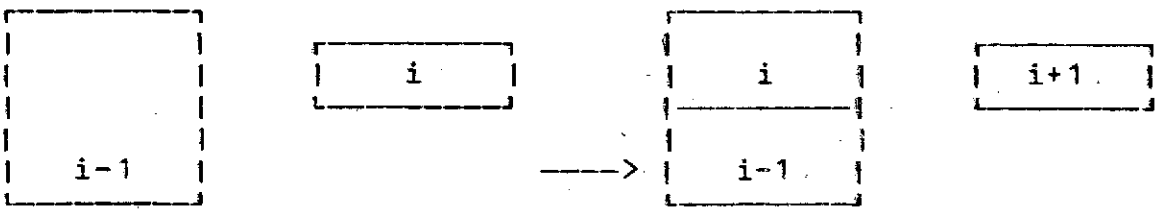

Figure 2

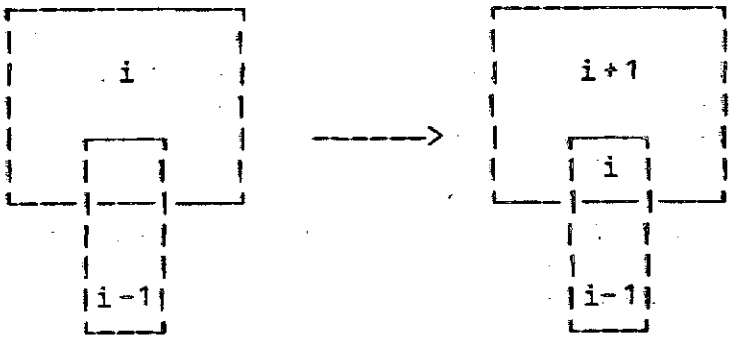

Figure 3

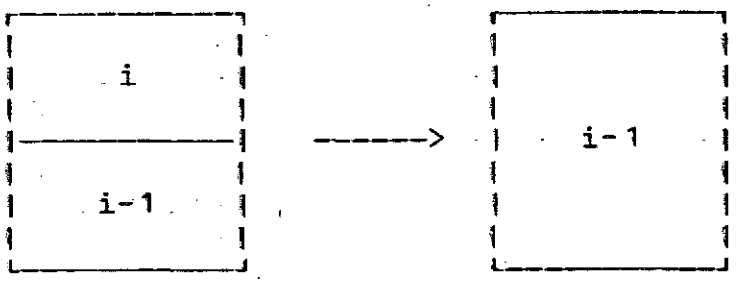

Figure 4

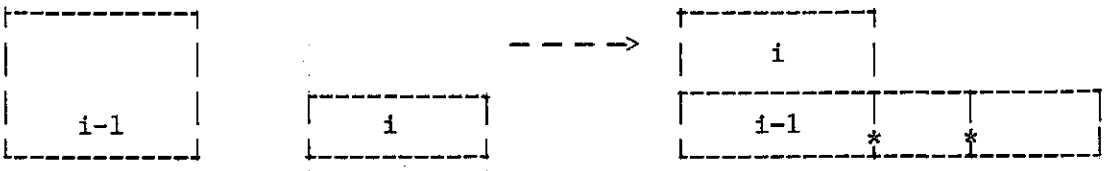

Figure 5

NMAP PROGRAM DESCRIPTION

F. Y. Borden

The primary purpose for which the NMAP program was designed is to assist the user in recognizing and visually correlating blocks of multispectral scanner remote sensor data on tape with areas seen on photographic imagery of the same flightline or scene. The user specifies the blocks on the tape to be mapped, the map symbols and class limits for classification, and the spectral bands (channels) to be used. The principal output consists of a map for the blocks specified according to the map symbols and class specifications used, with numbered scan lines and elements.

Computational Methods

The method is based on the norm of each observation in the data. An observation consists of the set of values for all channels for a single element in a scan line. For p channels, the observation for element j in scan line i can be represented as a p-valued vector,

$$X_{ij} = \begin{bmatrix} X_{ij1} \\ X_{ij2} \\ \vdots \\ X_{ijp} \end{bmatrix}$$

The norm of the observation $||X_{ij}||$ is then

$$||X_{ij}|| = \sqrt{X'_{ij} X_{ij}} \quad \text{or} \quad ||X_{ij}|| = \sqrt{\sum_{k=1}^{p} X_{ijk}^2}$$

Geometrically the norm is simply the length of the vector, X_{ij}, in p-dimensional space.

The norm of each observation is computed and transformed into the percentage of the maximum possible value for the norm. It is then translated into a mapping symbol of the class for which it falls within the class percentage limits.

The maximum possible norm value depends on the number of grey scale levels for each channel. For p channels and $n_1, n_2, \ldots, n_p$ grey scale levels in each of the p channels, the maximum possible norm value would be

$$\left[\sum_{i=1}^{p} (n_i - 1)^2 \right]^{1/2}$$

In most cases the number of levels is 64, 128, or 256.

Use of the Program

NMAP uses original or subset tapes in the format defined in the manual, "Digital Aircraft Multispectral Data Tape Format." Control cards are used to specify the flightline or scene (tape file name) which is to be used, to specify the blocks to be mapped according to scan line and element designations, and to specify mapping symbols and limiting percentages for classes to be used. Default options which are appropriate in most situations minimize control card preparation. Output consists of a title page with the control specifications for the run, map pages for each block requested, a summary of the classification results for each block, and a table of the frequency distribution by one percentiles for each block.

UMAP PROGRAM DESCRIPTION

F. Y. Borden

UMAP is useful for identifying areas of uniformity and non-uniformity in the remote sensor data by mapping. Such maps are valuable in the identification of suitably uniform areas for use as training fields for other analytical programs. During intermediate stages of analyses, the maps are useful as guidelines in judging the adequacy of maps from clustering and classification analyses. An alternate use of the program is for the delineation of high contrasts and boundaries of contrasting areas.

Computational Methods

The absolute value of the Euclidean distance between the end-points of two vectors is D. Let X_1 be the vector $\begin{bmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{1p} \end{bmatrix}$ and X_2 be $\begin{bmatrix} X_{21} \\ X_{22} \\ \vdots \\ X_{2p} \end{bmatrix}$. Geometrically these define

vectors with a common beginning point at the origin and X_1 and X_2 as the end-points in p-dimensional space. In remote sensor data, each vector is composed of the set of responses for the spectral bands defined by the multispectral scanner used to obtain the data. The squared distance between the two end-points, D^2, is found as $(X_1 - X_2)' (X_1 - X_2)$ or $\sum_{i=1}^{p} (X_{1i} - X_{2i})^2$. If D is small for a pair of vectors, the vectors are geometrically close together and numerically similar. A large value of D indicates a large contrast between the vector pair or a strong dissimilarity. Contrasts are computed in this way in UMAP and then translated into mapping symbols and mapped in the output.

The maximum value of a response in a channel depends on the number of grey scale levels for that channel. For p channels and $n_1, n_2, \ldots, n_p$ grey scale levels in each of the p channels, the absolute maximum and minimum values for D are $\left(\sum_{i=1}^{p} (n_i - 1)^2 \right)^{1/2}$ and 0, respectively.

Every point is identified by a scan line number and an element number within the scan line. Four D values are computed for each point, using as the other member of the pair for a D one of its near neighbors. Let the subscript i

designate the ith scan line and let j designate the jth element. The following D^2 values are computed:

$$D_1^2{}_{i,j} = \frac{1}{d_1} (X_{i,j} - X_{i,j+1})' (X_{i,j} - X_{i,j+1})$$

$$D_2^2{}_{i,j} = \frac{1}{d_2} (X_{i,j} - X_{i+1,j})' (X_{i,j} - X_{i+1,j})$$

$$D_3^2{}_{i,j} = \frac{1}{d_3} (X_{i,j} - X_{i+1,j+1})' (X_{i,j} - X_{i+1,j+1})$$

$$D_4^2{}_{i,j} = \frac{1}{d_3} (X_{i+1,j} - X_{i,j+1})' (X_{i+1,j} - X_{i,j+1})$$

The value D_{ij} is assigned the maximum $D_{k,i,j}$; $k = 1, 2, 3, 4$. The incremental spatial distance between the two elements of the pair is taken into consideration by d_1, d_2, and d_3. The reciprocal of these is the weighting value in a linear interpolation sense of the distance between the elements. In the case of every line and every element processing, d_1 is 1, the increment between two neighboring elements in the same line. Similarly, d_2 is 1 for the spatial increment between two elements in neighboring lines in the same position in each line. The value of d_3 is the hypotenuse value, $\sqrt{2}$, for two elements each on a line and an element position differing by one increment.

As detailed in a later section, it is possible to process, or have available for processing, data that are other than every line and every element. For data that are not every line and every element, d_1 would be the number of increments separating two neighboring elements on the same line, d_2 would be the number of increments separating elements on neighboring lines in the same element position, and d_3 would be $\sqrt{d_1^2 + d_2^2}$.

Values of D_{ij} are converted to symbols prior to mapping. First each d_{ij} is translated to a 0-100 scale by $D'_{ij} = 100\ (D_{ij} - D_{min})/(D_{max} - D_{min})$. The value D_{max} is $\left(\sum_{i=1}^{p} (n_i - 1)^2\right)^{1/2}$. D_{min} is 0 as described earlier. Each D'_{ij} is then assigned to the class within which limits it falls. The number of classes, the class limits, and the symbol for each class are under the control of the user. The symbol for the class within which the D'_{ij} falls is printed in position i, j of the output map. Class limits and their specifications are treated in more detail in a later section.

Use of the Program

UMAP has been written to use tapes with the format described in the manual, "Digital Aircraft Multispectral Data Tape Format." The input tape may contain a subset of the original data as processed by the SUBSET program. Control cards are used to do the following:

1. check the name of the specific input file to insure the correct file will be processed;
2. identify blocks of data to be processed;
3. define classes, class limits, and class symbols to be used; and
4. identify selected channels (spectral bands) of data to be used.

A control card naming the tape by its internal name is used if it is desired to check for the correct input tape. As many as 50 cards, each one specifying a block of data, may be used to select areas to be processed. Each block is defined by a beginning scan line, an ending scan line, a beginning element for all lines, and an ending element for all lines. The requested block must be contained within one of the tape file blocks. The specifications of the file blocks are obtainable from the SUBSET program output Table of Contents for the run that generated the tape or by use of

the TPINFO program. As an alternative option, all data in the file may be requested for processing. A line increment and an element increment may be specified on each block card. Reference should be made to the section on the function of line and element increments for a detailed description of their use.

One of three ways may be used for defining classes. A control card may be used for each class to be defined for as many as ten classes. The class control card will contain the upper limit of the class on a percentage basis, and the symbol to be mapped for map elements that have a D' value greater than the limit for the next lower class and less than or equal to the upper limit for this class. If the highest class limit is less than 100., each D'_{ij} that is greater than the highest specified upper limit will be printed as a blank.

The second means of defining classes is to input them on a control card that contains eight symbols, one for each class. In this case, the class limits are automatically set. The third means for class definition is by default. In this case, no control card is used and the following class definitions apply:

Class Limit	Class Symbol
3.	U
20.	⊭
100.	*

Output from the program consists of a title page with the general specifications for the run and a set of pages for each requested block giving the block specifications and the map of the block as well as a summary of the data. The summary contains the number of observations in each class, the overall average D value, and the maximum and minimum D values found in the data for the block. In addition, a table of the frequency distribution of D values by one percentiles is output. Blocks are sorted internally and, within each block, each line is mapped as it is encountered.

The Function of Line and Element Increments

The line increment and the element increment in a block specification determine the selection of lines and elements within lines that are processed within the limits of the specified block. They may be left unspecified on the control card in which case the corresponding increments for

the tape file block will be applied. It is necessary to spell out their function because their function does exert influence over the computation of the D values and the interpretation of the output.

In all cases except for the one described later in this section, only four elements are entered into the computation of a $D_{i,j}$ value. The four are taken from the corners of the spatial rectangle having the subscripts (i, j), $(i, j + m)$, $(i + k, j)$, and $(i + k, j + m)$ where k is the line increment and m is the element increment. Each of the four components of $D_{i,j}$ uses the appropriate weighting coefficient with $d_1 = m$, $d_2 = k$, and $d_3 = \sqrt{m^2 + k^2}$.

One special case exists for the UMAP program, that is when every line and element is available on tape for a requested block and where the line and element increments are specified by the user as 2. In this case, every other line and every other element in each line are output. However, instead of only four D values being compared, as discussed in the section on computational methods, 16 D values are compared with the largest one chosen for $D_{i,j}$. The 16 D values arise from four sets of four D values each. The value for $D_{i,j}$ is found by taking the maximum of the four following maximums: $D_{i,j}$, $D_{i,\,j+1}$, $D_{i+1,\,j}$, and $D_{i+1,\,j+1}$, where each of these has been found as the maximum of comparisons

according to the procedure presented in the computational methods section.

It should be pointed out that this is not the same as taking the maximum of the D values computed using only the four corner elements: $X_{i,j}$, $X_{i,j+2}$, $X_{i+2,j}$, and $X_{i+2,j+2}$.

Data Normalization

UMAP has the option for using unaltered data or normalized data. Normalized data is data that has been transformed in the following way. Let Z be a vector of normalized data. The normalization requirement is that $Z' Z = 1$, which can be accomplished by computing $Z = (X' X)^{-1/2} X$. This amounts to computing the sum of squares of the vector elements of X and then dividing each element by the square root of this value.

For unnormalized data, D as the distance between the end points of two vectors in p-dimensional space is influenced by the vector lengths as well as any angular separation between the vectors. A value of D computed with normalized data is influenced only by the angular separation between the two vectors. In normalized data the vectors

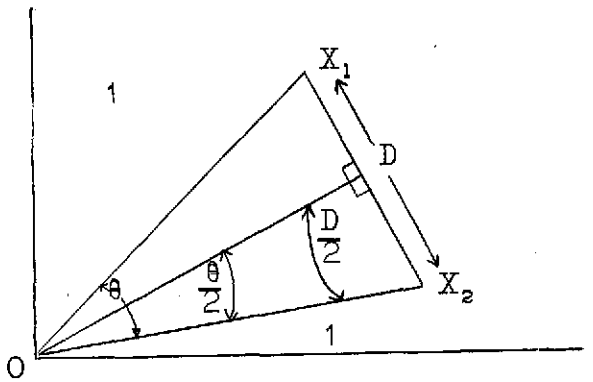

Figure 1

have unit length so that for the triangle formed by the origin and the vector end-points as shown in Figure 1, with θ as the angle of separation, $\sin \theta/2 = d/2$ and $\theta = 2 \arcsin (d/2)$. In normalized data then, only angular separation or the difference between relative reflectances are important.

If normalized data is to be used in a classification program, the training areas as identified using this program should be based on normalized data. If unnormalized data is to be used in a classification program, then this program should be run using unnormalized data.

STATS PROGRAM DESCRIPTION

H. M. Lachowski and F. Y. Borden

The purpose of the STATS program is to obtain basic statistical information for remote sensor data target areas within a flight path. These areas are frequently referred to as "training areas" and are used to estimate statistical parameters for specific targets. Training areas of any polygonal shape can be accommodated by STATS. A training area may be composed of distinct subareas for which the composite of these subareas would be processed as one unit. An area or a subarea is defined according to the coordinates of the corners on its perimeter and all of the data available within the area are used.

For each area, STATS computes and outputs the vector of means, the vector of standard deviations, and the variance-covariance matrix using all of the channels selected by the user. By option, it computes and outputs the correlation matrix, the frequency histograms for specified channels, and

the eigenvalues and eigenvectors for the variance-covariance or correlation matrices. In addition to printed output, optional output can be obtained in punch-card form, on magnetic tape, or in disk files.

It is recommended for efficiency in computer use that this program be used after the user has constructed a subset tape (described in SUBSET) that contains all of the parts of a flightline of interest to the user. In this way, the costly bypassing of unwanted data on the orignal tape can be avoided.

Computational Methods

Area Bounds

Each area for which statistics are to be computed is defined by the pairs of coordinates that designate the corners of a polygon. The coordinate pairs must be input in the order of their occurrence in either a clockwise or counterclockwise direction. The polygon need not be regular or convex but may be of any desired shape. It may have no more than forty sides, however. Two or more spatial areas (subareas) may be combined into one computation area (composite area), as long as the spatial areas do not

overlap and their combined number of sides do not exceed forty. In the following discussion a polygon defined by n pairs of coordinates will be considered. Let P_i be the ith point defined by the ith pair of coordinates. The coordinates are integers with the first being the scan line number and the second the element number in the scan line. The bounds of an area as defined by P_i, i = 1, ..., n, are translated to the beginning and ending element bounds for each scan line that appears in the area. This is done by computing the element for each scan line that is nearest to the line segment with end-points P_i and P_{i+1}. Each P_i, i = 1, 2, ..., n-1, is processed in this manner, finishing with the closing side using P_n and P_1. Since any shape polygon is allowed, it is possible for the boundary line to cross back and forth across one or more scan lines. In such cases, each affected scan line would have more than one pair of beginning and ending element values. As many as ten pairs of such values are allowed, which means the boundary line may cross a scan line as many as twenty times. This limitation is not likely to be important except in very unusual situations. A long winding path of a stream that is in a general direction parallel to the scan lines is one case where the boundary of the training area may cross some scan lines a number of times.

Statistics

Let B be the set of all pairs of coordinates in a bounded area. Consider X_{ij} as the vector $\begin{bmatrix} X_{ij1} \\ X_{ij2} \\ \vdots \\ X_{ijp} \end{bmatrix}$ for scan line i and element j composed of a response value for each of p channels. The mean vector and the variance-covariance matrix, C, are based on the X_{ij} for all (i, j) in B. The matrix C and $\overline{X}$ are computed in a typical and straightforward way and therefore not presented here. If the correlation matrix, R, is requested, it is computed in place of the C array area using the variance and covariance values.

The user has the option of having either or both the C and the R matrices output. Eigenvalues and eigenvectors may be computed and output based on either the C or the R matrices. The specifications for the eigenvalue-eigenvector computations are as follows, using Λ as the diagonal eigenvalue matrix, A as the eigenvector matrix, and the subscripts c and r as designators for the computations based on the C or R matrices, respectively:

$$C = A_c' \Lambda_c A_c; \qquad A_c' A_c = I$$

$$R = A_r' \Lambda_r A_r; \qquad A_r' A_r = I$$

except for a trivial case where $C = R$, $A_c \neq A_r$ and $\Lambda_c \neq \Lambda_r$.

Use of the Program

STATS uses original or subset tapes in the Digital Aircraft Multispectral Data Tape Format. The deck of control cards consists of a flightline name card that is optional followed by one or more sets of cards, each of which defines an area and the computations for the target for which the training area applies. The deck is completed by an END card. Control cards within each set of area cards are used to do the following:

1. specify whether unnormalized or normalized data is to be used,
2. specify the channels to be used,
3. request the eigenvalues and eigenvectors for either the variance-covariance matrix or the correlation matrix,

4. specify output options,
5. specify the name of the category for which the training area applies, and
6. specify the channels for which frequency histograms are to be computed and output.

Output consists of a title page with the control specifications for the run and, for each training area, the name of the category followed by the mean and standard deviation vectors, the variance-covariance matrix, and the other requested statistical information.

ACLASS PROGRAM DESCRIPTION

F. Y. Borden

ACLASS has as its purpose the classification and mapping of multispecral scanner remote sensor data. Each category is defined by a set of responses, one for each channel (spectral band). These data, with the category name and mapping symbol, form part of the input. Blocks of data to be classified and mapped are also specified by input. The data are classified according to their angle of separation, in a multidimensional geometric sense, from each of the categories, with the classification made into the category for which the angle is smallest. Each data unit is translated into the mapping symbol for the category to which it was assigned. Map output is made on a scan line by scan line basis for each specified block.

Additional output consists of auxiliary tables, one of which indicates the angles of separation between all pairs of categories specified. The program may be run for this information alone by deleting the map from computation.

Computational Methods

The classification of each selected element is made using its normalized vector and is based on its angle of separation from each of the normalized category vectors. An element in position j of scan line i will be represented as the vector $X_{ij} = \begin{bmatrix} X_{ij1} \\ X_{ij2} \\ \vdots \\ X_{ijp} \end{bmatrix}$ where the components of the vector are the responses in each of p channels (spectral bands). The equivalent normalized vector is Z_{ij}, which is computed as $Z_{ij} = (X'_{ij} \; X_{ij})^{-1/2} \; X_{ij}$. In other words, each

$$Z_{ijk} = X_{ijk} \Big/ \sqrt{\sum_{k=1}^{p} (X_{ijk})^2}$$

For the normalized vectors, $Z' \; Z = 1$. Let C_m be the normalized response vector for class m, m = 1, 2, ..., n, for the same p channels as in X_{ij}. The C_m vectors are input, having been estimated or established by means external to the program. Actually, the unnormalized C_m vectors can be input, since the program will normalize them. The reason for using normalized vectors instead of unnormalized vectors is that the assumption does not have to

be made that each C_m vector has the same magnitude of response in each channel as the X_{ij}, which, in fact, belong to category m. Furthermore, the assumption does not have to be made that any two X_{ij}, which are, in fact, in a single category, have the same magnitudes of response in each channel no matter on which side of the nadir each may occur and no matter where along the flightline each may be located. For example, for each C_m, the values may have been estimated from data from an entirely different flightline or from a laboratory spectral analysis. Using normalized vectors eliminates the need for the C_m vectors to be estimated from data from the flightline under analysis with its particular sun angle, general brightness, etc., which are characteristics more or less unique to the flightline.

The angle of separation between the vectors Z_{ij} and C_m is the criterion variable used for classification. The constraints are discussed later, but overlooking the constraints for the moment, Z_{ij} will be classified as belonging to category m if the angle between Z_{ij} and C_m is smaller than for any other C_ℓ, $\ell = 1, 2, \ldots, n$ and $\ell \neq m$. Any pair of vectors in p-dimensions define three points: the common origin and the end-point of each vector. The three points can be considered as a plane tiangle, as shown in Figure 1. As a result of normalization, two sides have unit

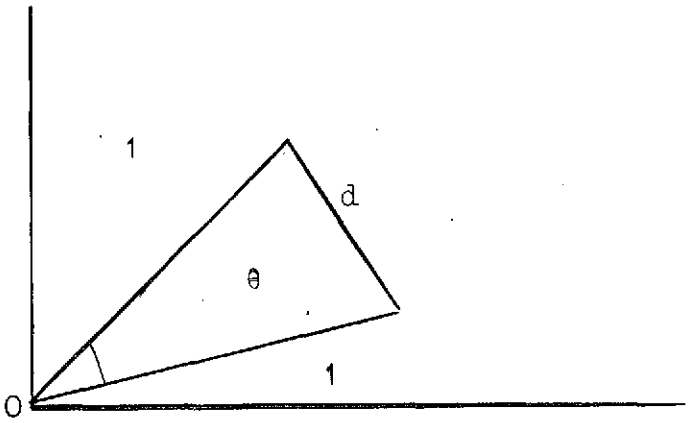

Figure 1

length, and d is the distance between the end-points. For the vector pair, Z_{ij} and C_m, let d_{ijm} be the distance between the end-points. The d_{ijm} are computable from

$$d_{ijm} = (Z_{ij} - C_m)' (Z_{ij} - C_m) = \sum_{k=1}^{p} (Z_{ijk} - C_m)^2$$

From Figure 1, $\sin(\theta/2) = d/2$ from which θ can be evaluated as $\theta = 2 \arcsin(d/2)$. Actually, in the program, these steps beyond the computation of d_{ijm}^2 do not take place inasmuch as only the smallest d_{ijm}^2 has to be found to identify the

value of m for which the smallest θ exists. This increases computational efficiency.

Constraints exist for the classification and these are in the form of a maximum allowable angle of separation, A_m, between Z_{ij} and C_m. If θ_{ijm} is greater than A_m, Z_{ij} will not be classified as belonging to category m. The angular values of A_m, m = 1, 2, ..., n, for n classes are input or set by default and are actually converted to critical distances so that the θ's do not have to be computed for the reason given above. The classification criterion is applied only to classes for which the above constraint is not violated. If the constraints are not met for any of the classes, then the observation is classified as "other." Since the A_m, m = 1, 2, ..., n, do not have to have the same value, it is possible that a Z_{ij} could be classified as belonging to class m for which the d_{ijm}, overlooking the constraints, is not the smallest. This could occur in cases where the A_m was large compared to the A_ℓ, for class ℓ for which the $d_{ij\ell}$ was the smallest but not small enough to meet the constraint imposed by A_ℓ.

To improve computational efficiency, since the program is not limited entirely by tape processing time, the following scheme was programed. Let b be the smallest distance of separation for all d_{gh}, g = 1, 2, ..., n and

$h = 1, 2, \ldots, (g-1)$, where $d_{gh} = (C_g - C_h)' (C_g - C_h)$. If, in the classification process for Z_{ij}, a class is encountered for which $d_{ijm} \leq b/2$ and d_{ij} does not violate the constraint for class m, Z_{ij} can be assigned to class m forthwith with no further investigation of other classes. In addition, if the first class, m, to which Z_{ij} is compared is the one for which $Z_{i,j-1}$ was assigned, it is most probable that Z_{ij} will be assigned to class m. The result would be that only one comparison would be made in the majority of cases instead of n, the number of classes, for each Z_{ij} processed. The program was constructed in this way because, in this kind of data, the probability that neighboring observations are of the same target is very high. This programing feature improved running time to nearly tape speed.

Once an observation has been assigned to a category, the map symbol is assigned to the position. After a scan line is completed, the line of map symbols is output.

Use of the Program

ACLASS uses original or subset data tapes in the Digital Aircraft Multispectral Scanner Data Tape Format.

Control cards are used to specify the flightline (tape file name) that is to be processed, the channels to be used, and the blocks to be classified and mapped according to scan line and element designations. In addition, they are the means of specifying the spectral characteristics of each of the categories, the category mapping symbol for each category, and the angular limit for each category within which an unknown must fall to qualify for consideration as belonging to the category. The output consists of a title page with the general specifications, including the table of angular separations for all pairs of categories. Map pages are output for each block and a summary table is output of the classification results by number and percentage in each category. A control card allows the user to choose not to have the map output, which saves substantially on run time since the remote-sensor data do not have to be processed in this case. This option is useful when the primary interest for the run is in the angular separation of pairs of categories and in the comparison of the spectral characteristics of the categories.

DCLASS PROGRAM DESCRIPTION

F. Y. Borden

This program works exactly like ACLASS, except that it does not normalize the signature vectors and, therefore must use the distance between any two vectors instead of the angle.

ACLUS PROGRAM DESCRIPTION

B. J. Turner

The purpose of ACLUS is the unsupervised classification and digital mapping of multispectral remote sensor data. It differs from ACLASS in that the user is not required to specify a set of spectral signatures initially. ACLUS develops its own set of spectral signatures using a clustering algorithm and outputs a map on the basis of these. The intensity of clustering and the intensity of sampling of the data to form these clusters are under user control.

Computational Methods

Remote sensor data is supplied to the program on digital magnetic tape in standard format. The user specifies by a control deck of cards or teletypewriter records: (a) the corner coordinates of the block(s) to be processed, (b) the number of sample points to be initially

chosen, and (c) the initial critical clustering angle, which is discussed later.

The program uses the block specifications to randomly select the required number of sample points, storing the coordinates in an array. These are then sorted by scan line number and element number within lines. If there are multiple blocks, then the number of sample points is allocated to each block in proportion to its size.

The clustering algorithm developed for ACLUS was influenced by a method suggested by Tryon and Bailey[1] as being useful when the number of observations is very large. The first stage of this method, which they called "iterative condensation on centroids," requires that trial group centroids be set up and each point is assigned to that group with which it has its smallest euclidean distance. After all have been assigned, the centroid coordinates are computed and the process iterated until no change in allocation occurs.

In the ACLUS program the initial centroids are computed from the first scan line in the specified block and from the

---

[1]R. C. Tyron and D. E. Bailey. 1972. Cluster Analysis. McGraw-Hill, New York, N. Y. pp. 147-150.

user-supplied initial critical angle, θ_c. If the vector of spectral data for the jth element within the ith scan line is designated as X_{ij} and its normalized analogue as Z_{ij} where $Z_{ij} = X_{ij}(X'_{ij} X_{ij})^{-1/2}$, then conceptually the procedure is as follows. The angle, θ, subtended at the origin in p-dimensional space (assuming each observational vector Z_{ij} has p elements) by Z_{11} and Z_{12} is computed. If this is less than θ_c, then the mean vector, C_i, is calculated and this becomes the first centroid. If $\theta > \theta_c$, then $C_1 = Z_{11}$ and $C_2 = Z_{12}$. Then Z_{13} is attached to whichever centroid with which it makes the smaller angle, unless the angle is greater than θ_c in which case a third centroid is formed. The centroid is recomputed with each additional observation, and a "moving" angular standard deviation is also computed. This procedure is carried out for every element in the first scan line. This defines the set of initial or trial centroids on which the sample points are to be "condensed." It can be seen that the number of initial centroids is controlled by the initial critical angle: the larger the angle, the smaller the number of initial centroids.

Each sample point is then located in turn on the data tape and is attached to the nearest centroid unless it deviates from this by an angle greater than some multiple of the angular standard deviation, in which case, it will form

a new centroid. If the point is accepted into an existing cluster, the mean vector and angular standard deviation are adjusted, and immediately adjacent points to the left and right along the scan line are tested to see if they are within the same cluster. If so, they are accepted and the centroid statistics are recomputed; if not, the next sample point is located. This technique makes use of the fact that there is a high probability that immediately adjacent observational points are spectral measurements of similar objects because of the spacial pattern relationships that exist in these data. The effect is to considerably augment the sample size at little additional computational cost.

After all sample points and their neighbors have been allocated, clusters that are represented by only one sample point are dropped. The remaining clusters are then tested to find if any overlap by one standard deviation. If so, the overlapping clusters are fused into one. The clusters are then sorted in descending order of their sample size. Clusters that have been formed from only a few sample points can be dropped (the user can specify the minimum proportional sample size for a cluster), and if there are still more than ten clusters remaining, the least represented clusters are dropped until only ten remain.

If the user now wishes to obtain a digital map of the same area, the tape is rewound and a character is assigned to each cluster spectral signature. Each observational element is assigned the character of the nearest cluster unless it is outside any cluster by some user-supplied multiple of the angular standard deviation in which case it is assigned a blank character. The matrix of characters so formed is printed out as a digital map.

Use of the Program

ACLUS uses original or subset data tapes in the Digital Aircraft Multispectral Scanner Data Tape Format. Control cards are used to specify the flightline or scene (tape file name) that is to be processed, the channels to be used, and the blocks to be classified and mapped according to scan line and element designations. In addition, they are used to specify the number of sample points to be selected, the initial critical angle for clustering, and the angular standard deviation multiplier used in the element-by-element classification for mapping. Options also exist for deleting low-reliability clusters, for obtaining extended output,

for obtaining the map only, and for deleting the mapping phase.

The following hints are offered for the benefit of new users.

1. Run the program first with all default options. To do this, input only the BLOCK card(s).
2. Compare the output map with the photographic imagery. For more detailed mapping, reduce the initial critical angle. For less detailed mapping, increase the critical angle or delete the low-reliability clusters.
3. To classify more of the mapped area (i.e., reduce the unclassified blank area), either (a) increase the initial critical angle or (b) increase the standard deviation multiplier. To obtain more blank unclassified area, either (a) decrease the initial critical angle, (b) decrease the standard deviation multiplier, or (c) delete the low-reliability clusters.
4. Adjust the sampling intensity if the size of the test block is very large or very small.
5. Vary only one factor at a time so that the effect is not confounded.

6. Generally, the initial critical angle should be in the range 1° to 10° and the standard deviation multiplier between 2 and 5. The number of sample points cannot exceed 900.

CANAL PROGRAM DESCRIPTION

H. M. Lachowski and F. Y. Borden

The CANAL program computes the canonical analysis for categories of multispectral scanner data based on the mean vectors and covariance matrices for the categories. The categories are defined and their basic multivariate statistics are obtained prior to the use of this program, for example, by the use of training areas and the STATS program. Each category is defined by a mean vector composed of the set of averages, one for each spectral band, and the corresponding covariance matrix. These basic statistics, in addition to the category names and mapping symbols, form the main part of the input to CANAL.

In the first part of the program, a canonical analysis is performed on the data for all categories. The minimum number of linear transformations yielding the maximum separability among the categories is obtained as a result of the canonical analysis. In the second part of the program,

each observation is first transformed using the linear transformations and then classified according to its euclidean distance of separation (in a multidimensional geometric sense) from the transformed mean vector of each of the categories. Classification is made into the category for which the distance is smallest if the distance is within a specified limit. If the distance exceeds the limit, the observation remains unclassified. The observation is then translated into the mapping symbol for the category to which it was assigned. A map of the classification results is output.

Additional output consists of auxiliary tables showing various matrices computed in the canonical analysis and distances of separation between all pairs of categories. The program may be run for the statistical information alone by optional termination of the program prior to the classification and mapping computations.

Use of the Program

The canonical analysis program uses original or subset tapes in the Digital Aircraft Multispectral Data Tape Format. Control cards are used to: (1) input the

specifications for each category, i.e., the number of observations, mean vector, covariance matrix, and the category name; (2) specify the flightline or tape file name; (3) specify the channels to be used and the blocks of data to be classified and mapped; (4) set category mapping symbols and limits; and (5) set various processing options.

Default options, which are appropriate in many situations, minimize control card preparation. Output consists of two parts: the canonical analysis and the classification and mapping. Part one consists of a title page with the control specifications for the run. The most important output of this part is the transformation matrix and the canonical axes that are used as the new signatures for the given categories. The output also contains various matrices computed in the canonical analysis. Part two consists of the table of separations for all pairs of categories and map pages for each block requested. For each block, a summary table is output of the classification results by number and percentage in each category. A control card allows the user to terminate the program before the classification and mapping is performed, thus saving substantially on run time. This option is useful when the primary interest for the run is the canonical analysis and the table with separations of pairs of categories.

Computational Methods

Canonical Analysis

Consider several p-variate universes, say h in number. Each universe may be conceived of as a swarm of points in p-dimensional space centered at a point characterized by a vector μ and dispersed about this point in an ellipsoidal pattern characterized by the covariance matrix Σ. The universes under consideration overlap to a greater or lesser degree and the mean vectors are more or less distinctly separated. A finite sample of observations can be obtained from each of the h p-variate universes. Since canonical analysis was explored for its potential use in the analysis of multispectral scanner data, it will be presented in this framework. Each sample of observations corresponds to a training set for a given category (target). Each training set, which is defined by the investigator, is chosen to be a representative sample of data for a homogeneous target; i.e., a target that has uniform characteristics differing from point to point within the target area only by random variability. Multispectral scanner measurements are taken as the exemplification of the uniform characteristics.

Each observation will be represented as a p-component vector,

$$X_{ij} = \begin{bmatrix} X_{ij1} \\ X_{ij2} \\ \vdots \\ X_{ijp} \end{bmatrix}$$

where p is the number of channels for element j in scan line i. The sample mean vector for the kth category (k = 1, 2, ..., p) is

$$\overline{X}_k = \frac{\sum_{\text{all } ij} \delta_{ij} X_{ij}}{n_k}$$

where $n_k = \sum_{\text{all } ij} \delta_{ij}$ for $\delta_{ij} = 1$ if element j in scan line i belongs to the training area for category k, and $\delta_{ij} = 0$ if element j in scan line i does not belong to the training area k, category k.

The sample covariance matrix for the kth category is

$$\hat{\Sigma}_k = \sum_{\text{all } ij} (\delta_{ij} X_{ijk} - \overline{X}_k)(\delta_{ij} X_{ijk} - \overline{X}_k)'$$

In addition to this, $\overline{X}$ is defined as a p x h matrix of all the category means composed of all $\overline{X}_k$, k = 1, 2, ..., h, mean vectors as

$$\bar{X} = \begin{array}{c} \\ \text{ch. } 1 \\ \text{ch. } 2 \\ \vdots \\ \text{ch. } p \end{array} \begin{array}{c} \begin{array}{cccc} \text{cat. } 1 & \text{cat. } 2 & \cdots & \text{cat. } h \end{array} \\ \begin{bmatrix} \bar{X}_{11} & \bar{X}_{12} & \cdots & \bar{X}_{1h} \\ \bar{X}_{21} & \bar{X}_{22} & \cdots & \bar{X}_{2h} \\ \vdots & \vdots & & \vdots \\ \bar{X}_{p1} & \bar{X}_{p2} & \cdots & \bar{X}_{ph} \end{bmatrix} \end{array}$$

The annotation of the above matrix indicates the category and channel organization of the matrix. Let N and n be, respectively, an h x h matrix and an h x 1 vector of the number of observations in the categories as

$$N = \begin{bmatrix} n_1 & 0 & \cdots & 0 \\ 0 & n_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & n_h \end{bmatrix}, \qquad n = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_h \end{bmatrix}$$

The method as presented here is based on the method by Bartlett[1] and by Seal[2]. It differs, however, from

---

[1]M. S. Bartlett. 1938. "Further aspects of the theory of multiple regression." Proceedings of the Cambridge Philanthropic Society, 34.

[2]H. L. Seal. 1964. Multivariate Statistical Analysis for Biologists. Methuen and Co., Ltd., London.

their presentations after the initial steps and is directed toward remote sensing data processing and analysis. The introductory details will not be repeated here since readers may refer to Seal (1964) for background information.

The objective of canonical analysis is to derive a linear transformation that will emphasize the differences among the sample estimates of the means of the h universes. In other words, the objective is to define new coordinate axes in directions of high information content useful for classification purposes.

The desired transformation for the general X and Y is

$$Y = CX$$

where C is the q x p transformation matrix where $q \leq p$, and Y is the transformed q-element observation vector. For every X_{ij} referenced by scan line i, element j, $Y_{ij} = CX_{ij}$ with

$$Y_{ij} = \begin{bmatrix} Y_{ij1} \\ Y_{ij2} \\ \vdots \\ Y_{ijq} \end{bmatrix}$$

The reason for $q \leq p$ is explained later.

Let W be the combined covariance matrix for all the categories; commonly referred to as the "within" category covariance matrix and computed as

$$W = \left(\sum_{i=1}^{h} n_i - h \right)^{-1} \left[\sum_{i=1}^{h} (n_i - 1) \hat{\Sigma}_i \right]$$

where $\hat{\Sigma}_i$ is the covariance matrix for category i, n_i is the number of observations for category i, and h is the number of categories. Let P be the "among" categories covariance matrix defined as

$$P = \bar{X}\,\bar{X}'\,N - \frac{1}{\sum_{i=1}^{h} n_i} (\bar{X}\,n)(\bar{X}\,n)'$$

In order to meet the objective of finding C so that the differences among the groups are emphasized, CPC' must be maximized since CPC' will be the "among" covariance matrix for the transformed variables, Y. The matrix C can only be made to be unique if additional constraints are placed on it. The constraint that CWC' = I, for I the q x q identity matrix, will suffice and, in addition, has the highly

desirable effect that under this constraint the transformed variables will be independent and have unit variances.

Maximization of CPC' subject to $CWC' = I$ cannot be solved directly; therefore, it has to be cast into a different form. This is accomplished in the following manner. Defining $W^{1/2}$ so that $W^{1/2} \cdot W^{1/2} = W$ and using $W^{1/2}W^{-1/2} = I$ and $(CW^{1/2})' = W^{1/2}C'$ then

$$CPC' = CW^{1/2}W^{-1/2}PW^{-1/2} \cdot W^{1/2} \cdot C'$$

$$= (CW^{1/2})W^{-1/2}PW^{-1/2}(CW^{1/2})'$$

In addition, $CWC' = I$ may be written as $(CW^{1/2'})(W^{1/2}C') = I$. Let $F = CW^{1/2'}$, then $FF' = I$. Let $V = W^{-1/2}PW^{-1/2'}$; then, by substitution of F and V, the problem is to maximize FVF' subject to $FF' = I$. This form is now a straightforward eigenvalue problem for which the only remaining difficulty is in finding $W^{1/2}$.

In order to obtain $W^{1/2}$, find A and Λ such that $A\Lambda A' = W$ constrained by $AA' = I$ where Λ is a diagonal matrix of eigenvalues extracted from W, and A is a matrix of corresponding eigenvectors. From this, $W^{1/2} = \Lambda^{1/2}A'$. Furthermore, $W^{-1/2} = A\Lambda^{-1/2}$.

Once $W^{-1/2}$ has been computed, the product $T = W^{-1/2}PW^{-1/2}$, can be obtained. The next next step is to find Z and F such that

$$W^{-1/2}PW^{-1/2}' = FZF'$$

and

$$FF' = I$$

where Z is the diagonal matrix of the eigenvalues of T, and F is a matrix of corresponding eigenvectors. These matrices are as follows:

$$Z = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{bmatrix}, \quad F = [f_1, f_2, \ldots, f_p]$$

The p eigenvalues of T are only distinguishable when $p \leq h - 1$. In the case when $p > h - 1$, there are $p - h - 1$ zero eigenvalues (or computational approximations of zero values) and $h - 1$ distinguishable non-zero eigenvalues. A suitable procedure to test whether all the eigenvalues after the qth can be ignored because they

are computational approximations of zero is the Bartlett's test.[1] Bartlett's test is based on the fact that

$$\left\{\left(\sum_{i=1}^{h} n_i - 1\right) - (p + h)/2\right\} \ln \prod_{j=q+1}^{m} (1 + \lambda_j)$$

is approximately a chi-square variable with (p - q) times (h - q - 1) degrees of freedom when $\lambda_{q+1} = \lambda_{q+2} = \ldots = \lambda_m = 0$. Here m is the smaller of h - 1 and p. This is accomplished by the testing of successive λ's for the given condition and stopping when the condition is satisfied.

Following this, Z is partitioned into q by q and p-q by p-q submatrices. The q by q partition,

$$Z^* = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_q \end{bmatrix}$$

contains the distinguishable eigenvalues and will be used as a discriminant space. In a similar manner, F is partitioned,

---

[1] M. S. Bartlett. 1947. "Multivariate analysis." Journal Royal Statistical Society, Supplement 9.

$$F^* = [f_1, f_2, \ldots, f_q]$$

As a result of partitioning Z and using a reduced discriminant space, only a certain portion of the total variance will be retained. The percentage of variance retained may be found from the following equation:

$$T = 100 \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

The equation for FZF' (page 10) now becomes

$$F^* Z^* F^{*\prime} \cong W^{-1/2} p W^{-1/2\prime}$$

and

$$F^* F^{*\prime} = I$$

The transformation matrix C may be computed from the equation for F (page 10), which becomes $F^* = C^* W^{1/2\prime}$. Therefore, C^*, which is now a q x p matrix, is computed as $C^* = F^* W^{-1/2}$.

As mentioned before, the possible rank q of the discriminant subspace depends on the relative sizes of p, the number of elements in the X vector, and on h, the number of categories. If h - 1 is less than p, then h - 1 is the maximum possible rank of the discriminant space. For example, if two classes are used, their centroids will have to fit on a single line, the centroids of three classes will have to fit on a plane, four classes in a three dimensional space, and so forth. If, however, $h - 1 \geq p$, it is possible to have as many as p canonical axes. If the smaller of p and h - 1 is quite large, one might decide to use less than the maximum number of axes for reasons of parsimony. In most cases, a reduced rank discriminant space yields adequate results when employed in classification. The ultimate aim is to reduce the problem of distinguishing between multivariate populations to the scale of a single variable.

Classification Procedure

The classification method used here is based on comparison of the euclidean distance between the input observation (unknown to be classified) and the stored references. Before the actual comparison takes place, the input vector is centralized; i.e., the grand mean is

substracted from it. It is then transformed according to the canonical transformation described in the previous section. The decision rule that was applied, based on the comparison of euclidean distances, is as follows: Y belongs to A if $|Y - \overline{Y}_i| < |Y - \overline{Y}_j| < C_i$; $j = 1, \ldots, h$; $i \neq j$. In this case, h categories are considered. Here C_i is the threshold value or limit for category i, $\overline{Y}_i$ is the sample mean for category i, and $\overline{Y}_j$, $j = 1, \ldots, h$, are the sample means for the remaining categories. This rule partitions the space into h + 1 regions (h categories plus "other"). The unknown observation is classified as belonging to category i if it is within the boundary limit defined by the threshold for category i, which is C_i. If it is outside all the h regions, the decision is made to classify the observation in the "other" or unclassified category.

The procedure is different if the threshold value is not used. In this case, an unknown observation is classified as belonging to the category for which the euclidean distance is smallest, without other limitations. Every observation, therefore, is classified as belonging to some category, but this does not necessarily mean that the decision is a correct one. The thresholds are used mainly to avoid classification in one of the h categories when the likelihood of success is marginal.

RATIO PROGRAM DESCRIPTION

F. Y. Borden

The RATIO program is a classification and mapping program for multispectral scanner remote sensor data based on the ratio of two selected channels (spectral bands). The program was designed primarily for vegetation analysis, therefore, the description is presented in this frame of reference. Using a general vegetation (or other) spectral signature specified by the user, data for each remote sensing unit that agree within a given tolerance to the signature are selected for ratio determination. For each remote sensing unit that is selected, the ratio of the two selected channels is computed and the remote sensing unit is assigned a mapping symbol corresponding to the class within which numerical boundaries the ratio value falls. For example, consider the two vegetation classes, coniferous and non-coniferous vegetation. It is well known that coniferous vegetation in general has less reflectance in the reflected infrared region than does non-coniferous vegetation. By

choosing the ratio denominator channel as one of those in the chlorophyll region and the numerator channel as one of those in the reflected infrared region, these two classes can be separated on the basis of the ratio. The ratio values for coniferous targets will be lower than those for the non-coniferous targets. The separation bounds for the targets must be specified by the user and can best be determined experimentally by using a sample of data from the scene to be analyzed.

In addition to map output, the frequency distribution table for ratio values is printed. This table is of particular value in choosing categories and in setting the bounds for the categories.

Computational Methods

The data for each element in each scan line is considered as a vector, say X_{ij} for scan line i, element j. For p channels, X_{ij} is composed as $\begin{bmatrix} X_{ij1} \\ X_{ij2} \\ \vdots \\ X_{ijp} \end{bmatrix}$. Let Z_{ij} be the normalized analog of X_{ij} so that $Z_{ij} = X_{ij}(X'_{ij}\, X_{ij})^{-1/2}$ and $Z'_{ij}\, Z_{ij} = 1$. Let V be a p-element vector, the values

for which are specified by the user to be the signature for, say, vegetation and let W be the normalized analog of V. The user determines whether normalized or unnormalized data will be used. This option is only effective in the screening of the X_{ij} prior to the ratio computation. If the unnormalized option is selected, the X_{ij} are screened by selecting for the ratio computation only those X_{ij} that have $d_{ij}^2 = (X_{ij} - V)'(X_{ij} - V) \leq D^2$, where D is the critical distance set by the user. The d_{ij} is the euclidean distance between the two points X_{ij} and V in p-dimensional space. For the unnormalized option, Z_{ij} are selected for ratio computation if $t_{ij}^2 = (Z_{ij} - W)'(Z_{ij} - W) \leq T^2$, where $T = 1/2 \arcsin(\theta/2)$ and θ is the critical angle set by the user. The d_{ij} and t_{ij} are related geometrically in that t_{ij} is directly related only to the angular separation of X_{ij} and V whereas d_{ij} is composed both of the angular separation and the vector lengths of X_{ij} and V. All data that are screened out are assigned blanks as mapping symbols.

For the selected data, the ratio $R_{ij} = X_{ijk}/X_{ij\ell}$ is computed using channels k and ℓ as designated by the user. The selection of the normalized or unnormalized data option has no influence on the ratio since $X_{ijk}/X_{ij\ell} = Z_{ijk}/Z_{ij\ell}$. Let B_n, $n = 1, \ldots, m$, be the upper bounds in ascending order for the ratio in classifying the ratios into m

categories; $B_0 = 0$. Then if $R_{i,j} > B_n$ for n = 1, ..., k-1 and $R_{i,j} \leq B_k$, the mapping symbol for class k is assigned to the element. If $R_{i,j} > B_m$, the mapping symbol for the last class, class m, is assigned. The user defines the bounds by input.

The frequency distribution table of ratio values that is output is computed based on minimum and maximum ratio values specified by the user. For this table, one hundred equally spaced classes are set, with the minimum ratio as the lower bound of the first class and the maximum ratio as the upper bound for the last class. Values that are below the lowest bound or above the highest bound are assigned to the first or last class respectively. The frequency distribution output is valuable in setting the number of ratio categories and their bounds.

MERGE PROGRAM DESCRIPTION

D. N. Applegate and F. Y. Borden

The MERGE program is used to merge satellite data from two ORSER formatted data tapes, each tape containing one or more passes of the same area and each being from a different date. The final merged tape may contain up to six different passes. These merged data tapes are useful in studying the effects of temporal change and to perhaps improve classification of certain targets.

Tape Formats

Every tape generated by the MERGE program has the same tape format as the original data tapes. Every merged tape can be used by any of the programs that can operate on the original data tapes. The identification record of the first source tape is reproduced on the merged tape, except that the channels from the second source tape are added to the

channels from the first tape. The table of contents for the merged tape corresponds to the BLOCK data card described in the control card section. All lines and elements on the merged tape are numbered as they are on the first source tape.

Use of the Program

The two source tapes may be merged tapes themselves, but both must contain the area to be merged. The block to be merged from the first source tape must be specified as input to the program. Line and element differences from the first source tape to the second tape must also be input; these values are used to compute the block to be merged from the second source tape. One way to calculate these differences is to overlay digital maps from each tape; there should be no rotational effect evident. Channels from the source tapes are renamed to avoid duplicate channel numbers. Program output consists of tape information pages for each of the two source tapes and one for the merged tape.

MAPCOMP PROGRAM DESCRIPTION

D. N. Applegate and F. Y. Borden

The MAPCOMP program is a program that compares, element by element, two digital classification maps of the same ground area. The program was designed especially to compare maps generated from merged satellite data. By using the MAPCOMP program, one can compare classification results from two different passes of the same area, each pass being from a different date; or one can study the results of classification using selected channels from different dates.

Each map is classified, line by line, according to the computational methods described in the ACLASS (DCLASS) program description. Each element from each map is assigned a category symbol or a category number, depending on which option was specified in the control cards. The elements are then compared and a symbol assigned designating whether the elements were equal, the elements were not equal, the first element equaled a blank or the "other" category and the second did not, the second element equaled a blank or the

"other" category and the first did not, or both elements equaled a blank or the "other" category.

Use of the Program

Input to the program is any tape in the ORSER format, but merged tapes are primarily used. Channels to be used for each map, and the limiting distances or angles for each map for each category, should be specified in the input to the program, or the default values will be assigned. One set of category cards apply to both maps. However, a separate set of signature cards may be input for each map. Brightness factors may be specified for each set of channels by the use of the NORM cards described in the control card section below.

Program output consists of a title page giving the channels used in the classification of each map, and a list of the category names, symbols (if used), limits, and category signatures for each map. A comparison map and a

summary table listing the five cases noted in the introduction above, with corresponding counts and percentages, are output.

PRINCOM PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Principal Components Analysis (PRINCOM) program is to compute a transformation matrix from a set of observations within a chosen data site for use when performing principal components analysis. The rows of the transformation matrix correspond to the eigenvectors of the data site covariance matrix computed from eigenvalues arranged in descending order of magnitude. The data site mean vector and covariance matrix are acquired from the ORSER program STATS. The mean vector and transformation matrix are output into the BAT file* $PRNCOM for presentation to the classification programs when using principal components analysis as a preprocessing option.

The printed output consists of the following:

1. Echo-check of the mean vector and covariance matrix for the data site.
2. Eigenvalues of the covariance matrix.
3. Matrix with eigenvectors in columns.
4. Transformation matrix with eigenvectors in rows.
5. Resultant matrix computed by multiplying the transformation matrix by its transpose.
6. Percent of total variance represented by each vector in the transformation matrix and the cumulative percent variance.

Output on BAT file $PRNCOM consists of the following:

1. Data site mean vector
2. Transformation matrix.

---

*The batch and terminal (BAT) file receives output from remote job entry terminals. This data can later be retrieved as output or used as input to another program.

MINDIS PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Parametric Classifier with Linear Discriminant Function (MINDIS) program is to implement a minimum distance classifier based on pattern class means. The linear discriminant function used has the following form:

$$g^i(\underline{x}) = \underline{m}^{i\prime}\underline{x} - (1/2)\underline{m}^{i\prime}\underline{m}^i$$

A mean vector, $\underline{m}^i$, for each class is first computed from a set of training patterns and then a selected data site is classified using these computed means and the discriminant function shown above. Classification is completed by choosing the class which corresponds to the largest discriminant function.

The printed output consists of the following:

1. Echo-check of input control cards.
2. Echo-check of mean vector and transformation matrix, if principal components option is selected.
3. Patterns from each training block (optional).
4. Number of patterns in each class.
5. Sample training patterns from each class.

PARAM PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Parametric Classifier with Quadratic Discriminant Function (PARAM) program is to implement a classifier based on the statistical parameters of a training set of patterns from selected classes. The covariance matrix and mean vector for each pattern set is acquired from the ORSER program STATS.

When a pattern is input, a discriminant function is computed using the input parameters in the following form:

$$g^{i}(\underline{x}) = \ln p(i) - (1/2) \ln |\Sigma_i| - (1/2)[(\underline{x} - \underline{m}_i)'\Sigma_i^{-1}(\underline{x} - \underline{m}_i)]$$

where p(i) is the probability that a random pattern, $\underline{x}$, belongs to the i'th class. If the probabilities are unknown or otherwise omitted from the output, the program assumes the probability of each class to be equal.

A training set, i.e., blocks of patterns known to belong to certain classes, is input along with a selected data site. PARAM first classifies the training set and outputs the percent correct classification of each test block, to give an indication of classifier performance. The program then classifies and outputs a map of the data site, followed by a summary for each class. The PARAM program uses the theoretically optimal discriminant function for normally distributed patterns.

The printed output consists of the following:

1. Echo-check of input control cards.
2. Echo-check of mean vector and covariance matrices for each class.
3. Echo-check of mean vector and transformation matrix, if the principal components option is selected.

4. Transformed mean and covariance matrix for each class, if the principal components option is selected.

5. Inverse of and inverse times the covariance matrix for each class.

6. Patterns from each test block selected.

7. Percent correct classification of the test set.

8. Heading with flight line information.

9. Map of data site.

10. Block summary.

NPAR PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Nonparametric Trainer with Linear Discriminant Function (NPAR) program is to find a set of weights for use in a linear classifier with a discriminant function of the form:

$$g^i(\underline{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + w_{d+1}$$

Training sets of patterns belonging to selected classes are input and nonparametric training is accomplished using the fixed increment rule as the error correction procedure. Weights from the final training run are output to the BAT file* $WTS for input to the NPARMAP program.

The printed output consists of the following:

1. Echo-check of input control cards.
2. Echo-check of mean vector and transformation matrix, if the principal components option is selected.
3. Heading with flight line information.
4. Patterns from each training block selected.
5. Training run number, percent correct classification of the training set, and values of the weights after each training run.
6. Final classification of the training set after stopping rule was implemented.

Output to BAT file $WTS consists of the following:

1. Number of classes.
2. Final weights matrix (classes, channels)

---

*The batch and terminal (BAT) file receives output from remote job entry terminals. This data can later be retrieved as output or used as input to another program.

NPARMAP PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Nonparametric Classifier and Mapper with Linear Discriminant Function (NPARMAP) program is to classify and map a selected data site using a linear discriminant function of the form:

$$g^i(\underline{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + w_{d+1}$$

The number of classes and weights for each class are input from the NPAR program through the BAT file* $WTS. A pattern is classified into the class which yields the largest discriminant function resulting after the weight vector of each class is multiplied by the pattern.

The printed output consists of the following:

1. Echo-check of input control cards.
2. Echo-check of number of classes and weights matrix.
3. Echo-check of mean vector and transformation matrix, if principal components option is selected.
4. Heading with flight line information.
5. Map of data site.
6. Block summary.

---

*The batch and terminal (BAT) file receives output from remote job entry terminals. This data can later be retrieved as output or used as input to another program.

QUADNPAR PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Nonparametric Trainer with Quadratic Discriminant Function (QUADNPAR) program is to find a set of weights for use in a classifier employing a quadratic discriminant function of the form:

$$g^i(\underline{x}) = \sum_{j=1}^{d} w_{jj} x_j^2 + \sum_{j=1}^{d-1} \sum_{k=j+1}^{d} w_{jk} x_j x_k + \sum_{j=1}^{d} w_j x_j + w_{d+1}$$

Training sets belonging to selected classes are input and the patterns are passed through a quadratic processor. Nonparametric training is conducted, using the fixed increment rule as the error correction technique. After each training run the training set is classified and the percent correct classification is output. Weights from the final training run are output to the BAT file* $QWTS for input to the QUADMAP program.

The printed output consists of the following:

1. Echo-check of input control cards.
2. Echo-check of mean vector and transformation matrix, if principal components option is selected.
3. Heading with flight line information.
4. Patterns from each training block.
5. Sample of quadratic patterns.
6. Training run number and percent correct classification of the training set.

---

*The batch and terminal (BAT) file receives output from remote job entry terminals. This data can later be retrieved as output or used as input to another program.

7. Final classification of the training set after the last training run.

8. Final weights and the average percent classification.

Output to BAT file $QWTS consists of the following:

1. Number of classes.

2. Indices for quadratic processor.

3. Final weights matrix (classes, indices).

QUADMAP PROGRAM DESCRIPTION

J. R. Hoosty

The purpose of the Nonparametric Classifier and Mapper with Quadratic Discriminant Function (QUADMAP) program is to classify and map a selected data site using weights from the program QUADNPAR and a quadratic discriminant function of the form:

$$g^i(\underline{x}) = \sum_{d=1}^{d} w_{jj} x_j^2 + \sum_{j=1}^{d-1} \sum_{k=1}^{d} w_{jk} x_j x_k + \sum_{j=1}^{d} w_j x_j + w_{d+1}$$

Each pattern in the data site is passed through a quadratic processor, and the resultant vector is multiplied by weight vectors from each class yielding a discriminant function. The number of classes, indices for the quadratic processor, and class weights are input from the QUADNPAR program through the BAT file* $QWTS. Classification is completed by choosing the largest of the discriminant functions.

The printed output consists of the following:

1. Echo-check of input control cards.
2. Echo-check of the number of classes, indices, and weights for each class.
3. Echo-check of mean vector and transformation matrix, if principal components option is selected.
4. Heading with flight line information.
5. Map of data site.
6. Block summary.

---

*The batch and terminal (BAT) file receives output from remote job entry terminals. This data can later be retrieved as output or used as input to another program.